REMARKS

I. Introduction

In response to the Office Action dated April 21, 2005, claims 1, 5, and 9 have been amended, and claims 13-15 have been added. Claims 1-3, 5-7, 9-11, and 13-15 remain in the application. Re-examination and re-consideration of the application, as amended, is requested.

II. Prior Art Rejections

In paragraphs (3)-(4) of the Office Action, claims 1-3, 5-7, and 9-11 were rejected under 35 U.S.C. §103(a) as being unpatentable over Kish et al., U.S. Patent No. 5,890,176 (Kish) in view of Baisley et al., U.S. Patent No. 6,625,663 (Baisley).

Specifically, the independent claims were rejected as follows:

> As to claims 1, 5, 9, Kish teaches "obtaining a request to save a file in a requested file version, wherein the file contains an object" at col. 2, lines 20-34, col. 9, lines 14-42, , col. 10, lines 51-53, Fig. 6;
>
> "determining if the requested file version is earlier than an object introduction version of the object" at col. 2, lines 20-34, col. 10, lines 42-58, Fig. 6;
>
> Ksih does not explicitly teach the following limitations. However, Baisley teaches "saving the file by streaming out data representing an instance of the object to the file in the requested file version if the requested file version is equal to or later than the object introduction version, wherein the data comprises actual methods and attributes of the object" at col. 6, line 51 to col. 7, line 24, Figs. 5A-5B;
>
> "saving the file by streaming out the data representing the instance of the object to the file, in the object introduction version if the requested file version is earlier than the object introduction version" at col. 6, line 51 to col. 7, line 24, Figs. 5A-5B.

Applicant traverses the above rejections for one or more of the following reasons:

(1) Neither Kish nor Baisley teach, disclose or suggest determining/comparing if a requested file version is earlier than an object introduction version of an object in a file to be saved;

(2) Neither Kish nor Baisley teach, disclose or suggest a request to save a file in a particular/requested file version;

(3) Neither Kish nor Baisley teach, disclose or suggest streaming out an object in a requested file version of the requested file version is equal to or later than the object introduction version; and

(4) Neither Kish nor Baisley teach, disclose or suggest streaming out an object in an object introduction version if the requested file version is earlier than the object introduction version.

Applications (such as word processing programs, spread sheet programs, or drawing programs) are used to create files (e.g., a documents, spreadsheets, or drawings) (see page 2, lines 15-21). The created files often contain objects (e.g., a drawing file may contain a line object, shape object, circle object, arc object, etc.) (see page 2, lines 19-21). When saving a file, the data for the objects within the file are streamed out with the file (see page 2, lines 21-22). In addition, the file is related to the version of the application that created the file. However, objects may also evolve and have different versions over time. In this regard, if the user desires to store a file in an earlier version than the object was first introduced, the evolution of the object may be hindered (see page 3, lines 1-7).

Independent claims 1, 5, and 9 are generally directed to storing object data in a particular version (see page 4, lines 8-12). The claims provide that the representative data (i.e., for the object) comprises actual methods and attributes of the object (see page 7, lines 20-21). The claims further provide for storing a particular version of the object by streaming data for a particular version of the object out to a file (see page 4, lines 8-12).

More specifically, the version of the file (that the object is being stored in) is compared to the version of the object when the object was introduced/originated (step 500 of FIG. 5, and page 10, lines 10-14). If the file version is the same or newer than when the object was first introduced, the file is saved such that (the data representing) the instance of the object is streamed out (e.g., stored) in the file version (step 506 and page 10, lines 15-20).

However, if the file version is older than when the object was introduced, the file is saved such that the (data representing) the instance of the object is streamed out (e.g., stored) in the object introduction version (see steps 502 and 504, page 10, line 21-page11, line 5).

Dependent claims 2, 6, and 10 further provide that if the file version is earlier than the object introduction version, the object is represented as a proxy object (when a file is opened), and when the file is saved, the proxy object holds onto the object's data and streams out the object's data (see steps 502 and 504 and page 11, lines 6-19).

Dependent claims 3, 7, and 11 provide the ability for superior objects to query the object to determine the appropriate version to stream out to the file (see step 306 and page 9, lines 15-21). The object responds to the superior object (that actually performs the streaming out operation) to stream out in the requested file version (if the requested file version is equal to or later than the object introduction version) and in the object introduction version (if the requested file version is earlier than the object introduction version) (see page 9, line 22-page 10, line 1; page 10, line 17-page 11, line 5).

New dependent claims 13-15 further provide that the application that is performing the save operation is a different version than the requested file version.

The cited references do not teach nor suggest these various elements of Applicant's independent claims.

The first element of the claims relate to a request to save a file in a requested/particular file version. It is also noted that the second claim element utilizes the requested file version that was part of the save request. Accordingly, before performing the determining operation, the request to save a file must be conducted. In rejecting this claim element, the Office Action relies on Kish col. 2, lines 20-34, col. 9, lines 14-42, col. 10, lines 52-53, and Fig. 6. Col. 1, lines 20-34 provide that if a document is changed, then an object version is compared to a document version currently being edited. If the versions are different, a save operation is conducted. Thus, rather than performing the determination after a save request has been received (as claimed), Kish first performs a determination and then performs a save operation depending on the result of the operation. The remaining cited portions of Kish are consistent with col. 2, lines 20-34. Nonetheless, such a comparison and sequence of steps described in Kish are not equivalent nor do they render obvious the present claims.

The second claim element is that of determining if the requested file version is earlier than an object introduction version of an object in a file. The Office Action rejects these claims based on Kish col. 2, lines 20-34, col. 10, lines 42-58, and Fig. 6. Col. 2, lines 2-34 explicitly conduct a comparison "to determine whether the object version is same as the document version currently being edited". Again, such a teaching is not even remotely similar to that set forth in the present claims. Firstly, a requested file version is not the same nor equivalent to a document version currently being edited. The requested file version may not be the same as the version of the document being currently edited. In fact, the new dependent claims 13-15 specifically provide that these versions are different. In this regard, the concept of the version of a document being currently edited has an entirely different connotation than a requested file version. Such terms refer to different stages in an application's process and to different elements/properties.

Secondly, the object introduction version is not equivalent to an object version. As described in Kish FIG. 6, all of the objects appear to have been introduced on or before version 0 of the application. However, the version used in the various comparisons as described in Kish are those identified at 618, 620, 622, and 624 – i.e., merely edits or changes to the object and not when the object was introduced.

In view of the above, the comparison conducted in Kish is not remotely equivalent to (nor does it render obvious) the comparison conducted in the present claims.

The independent claims continue and describe the process of saving the file by streaming the object based on the comparison. The claims provide for streaming out the data in particular versions depending on the comparison. In rejecting these claims, the Office Action admits that Kish fails to teach the claim elements and relies on Baisley instead. However, Baisley also fails to even remotely describe the claim limitations.

Firstly, as noted in the claims, the object is streamed out in either the object introduction version OR the requested file version depending on the comparison. However, in Baisley, all of the versions for a model are streamed out (see col. 4, lines 58-61; col. 5, lines 45-46) and not just one version as claimed.

Secondly, the streaming out operations relied upon in the Office Action are conducted if there is a non-derived, classifier-level structural feature that has not yet streamed. Such a result is not even remotely similar to performing the save operation based on a request to save a file followed by a comparison (as claimed).

Thirdly, Baisley does not even remotely describe streaming out a file in a requested file version based on a result of a specific comparison. Namely, Baisley does not describe, teach, or suggest, implicitly or explicitly, streaming out an object in a requested file version IF the requested file version is equal to or later than the object introduction version. In fact, Baisley doesn't even look to the comparison as set forth in the claims. As set forth in col. 6, lines 60-62, the value of a feature is compared to a start version of an object. In this regard, the value of a feature cannot and does not describe nor allude to a requested file version. In addition, rather than determining if the value is equal to or later than the start version, Baisley merely determines if they are different. If they are different (regardless of whether it is equal to or later than as claimed), a version attribute (in XMI) is written within an element (see col. 6, lines 62-65).

Baisley then continues and conducts another comparison to determine if there exists a version of the object in which the value (of the feature) was unset or replaced that is different from the end version of the object (see col. 6, line 65-col. 7, line 2). Again, nowhere in any of this text is there even a remote reference to a requested file version. Instead, this comparison merely looks to an end version of an object and NOT to an object introduction version of an object. Further, this portion of Baisley merely compares different object versions and not a requested file version to an object introduction version (as claimed).

In addition to the lack of teaching relating to saving the file in the requested file version, Baisley also fails to teach the streaming out in the object introduction version if the requested file version is earlier than the object introduction version. In this regard, Applicants direct the Examiner to the description of the comparisons conducted in Baisley described in the preceding paragraph.

The last element of the rejection of the independent claims states that the combination would enable Kish's users to create a single master document containing all versions without duplicating the entire document for each version and to maintain copies of all previous revisions so that all changes can be undone and redone. Regardless of whether the combination would enable such a teaching, such a result is not remotely similar to what the claims provide. The claims are not directed towards saving all versions in a single document. Instead, the claims explicitly teach away from such a teaching by receiving a request to save a file in a particular requested file version (and not all file versions) and then saving the file in the requested file version with the version of each object saved in the file changing (depending on the result of the comparison). However, nowhere in the specification or claims of the present invention is there any attempt to save multiple versions of a file in a single document.

Moreover, the various elements of Applicant's claimed invention together provide operational advantages over Kish and Baisley. In addition, Applicant's invention solves problems not recognized by Kish and Baisley.

Thus, Applicant submits that independent claims 1, 5, and 9 are allowable over Kish and Baisley. Further, dependent claims 2-3, 6-7, 10-11, and 13-15 are submitted to be allowable over Kish and Baisley in the same manner, because they are dependent on independent claims 1, 5, and 9, respectively, and thus contain all the limitations of the independent claims. In addition, dependent claims 2-3, 6-7, 10-11, and 13-15 recite additional novel elements not shown by Kish and Baisley.

III. Conclusion

In view of the above, it is submitted that this application is now in good order for allowance and such allowance is respectfully solicited. Should the Examiner believe minor matters still remain that can be resolved in a telephone interview, the Examiner is urged to call Applicant's undersigned attorney.

Respectfully submitted,

GATES & COOPER LLP
Attorneys for Applicant(s)

Howard Hughes Center
6701 Center Drive West, Suite 1050
Los Angeles, California 90045
(310) 641-8797

Date: July 21, 2005

By: Jason S. Feldman
Name: Jason S. Feldmar
Reg. No.: 39,187

JSF/bjs